BEVDet: High-Performance Multi-Camera 3D Object Detection in Bird-Eye-View

Junjie Huang [★], Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du

PhiGent Robotics
{junjie.huang,zhengzhu}@ieee.org,
{guan.huang, yun.ye, dalong.du}@phigent.ai

Abstract. Autonomous driving perceives its surroundings for decision making, which is one of the most complex scenarios in visual perception. The success of paradigm innovation in solving the 2D object detection task inspires us to seek an elegant, feasible, and scalable paradigm for fundamentally pushing the performance boundary in this area. To this end, we contribute the BEVDet paradigm in this paper. BEVDet performs 3D object detection in Bird-Eye-View (BEV), where most target values are defined and route planning can be handily performed. We merely reuse existing modules to build its framework but substantially develop its performance by constructing an exclusive data augmentation strategy and upgrading the Non-Maximum Suppression strategy. In the experiment, BEVDet offers an excellent trade-off between accuracy and time-efficiency. As a fast version, BEVDet-Tiny scores 31.2% mAP and 39.2% NDS on the nuScenes `val` set. It is comparable with FCOS3D, but requires just 11% computational budget of 215.3 GFLOPs and runs 9.2 times faster at 15.6 FPS. Another high-precision version dubbed BEVDet-Base scores 39.3% mAP and 47.2% NDS, significantly exceeding all published results. With a comparable inference speed, it surpasses FCOS3D by a large margin of +9.8% mAP and +10.0% NDS. The code will be released to facilitate future research.

Keywords: Computer Vision, Autonomous Driving, 3D Object Detection

1 Introduction

2D visual perception has witnessed rapid development in the past few years and emerged some outstanding paradigms like Mask R-CNN [13], which is high-performance, scalable [2, 4], and multi-task compatible. However, with respect to the scene of vision-based autonomous driving where both accuracy and time-efficiency are desired, major tasks like 3D object detection and map restoration (*i.e.*, Bird-Eye-View (BEV) semantic segmentation) are still conducted by different paradigms in the up-to-date benchmarks. For example, in the nuScenes [1] benchmark, image-view-based methods like FCOS3D [49] and PGD [50] have
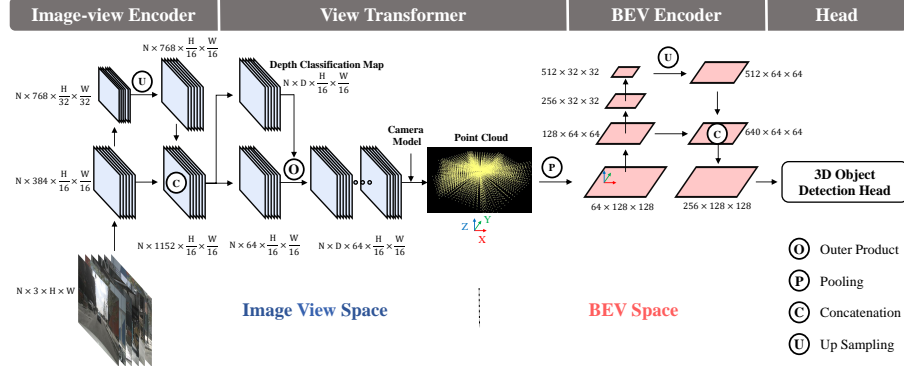

[★] Corresponding author.

Fig. 1. The framework of the proposed BEVDet paradigm. BEVDet with a modular design consists of four modules: Image-view encoder, including a backbone and a neck, is applied at first for image feature extraction. View transformer transforms the feature from the image view to BEV. BEV encoder further encodes the BEV features. Finally, a task-specific head is built upon the BVE features and predicts the target values of the 3D objects. We take BEVDet-Tiny as an example for illustrating the channels of different modules.

leading performances in the multi-camera 3D object detection track, while the BEV semantic segmentation track is dominated by the BEV-based methods like PON [39], Lift-Splat-Shoot [33], and VPN [31]. Which view space is more reasonable for perception in autonomous driving, and can we handle these tasks in a unified framework? Aiming at these questions, we propose BEVDet in this paper. With BEVDet, we explore the advantages of detecting 3D objects in BEV, expecting a superior performance compared to the latest image-view-based methods and a consistent paradigm with BEV semantic segmentation. In this way, we can further verify the feasibility of multi-task learning, which is meaningful for time-efficient inference.

The proposed BEVDet, as illustrated in Fig. 1, shares a similar framework with the up-to-date BEV semantic segmentation algorithms [33, 39, 54]. It is modularly designed with an image-view encoder for encoding features in image view, a view transformer for transforming the feature from image view into BEV, a BEV encoder for further encoding features in the BEV perspective, and a task-specific head for performing 3D object detection in the BEV space. Benefiting from this modular design, we can reuse a mass of existing works which have been proved effective in other areas and still expect a long way to upgrade this paradigm specific for the task of 3D object detection.

Though simple in constructing the framework of BEVDet, it is nontrivial to build its robust performance. When verifying the feasibility of BEVDet, for reasonable performance, the data processing strategy and the parameter number of BEVDet are set close to the image-view-based 3D object detector like FCOS3D [49] and PGD [50]. Unexpectedly, a serious over-fitting problem is ob-

served in the training process. Some cues reveal that the devil is in the excessive fitting capacity of BEVDet in the BEV space. First of all, the over-fitting encourages us to apply a complicated data augmentation strategy in the image view space as Lift-Splat-Shoot [33] for a regularization effect. However, this modification has a positive effect only if the BEV encoder is absent. Otherwise, it even degrades the performance. On the other hand, the batch size of the image-view encoder is N (*i.e.*, the numbers of cameras like 6 in nuScenes [1]) times that of the subsequence modules. Insufficient training data is also partly responsible for the over-fitting in BEV space learning. Besides, we observe that the view transformer connects the image view space with the BEV space in a pixel-wise manner, which decouples them from the perspective of data augmentation. This makes the data augmentation in image view have no regularization effect on the subsequence modules (*i.e.*, the BEV encoder and the 3D object detection head). Thus, as a supplement, additional data augmentation operations are conducted in the BEV space like flipping, scaling, and rotating for the model's robustness on these aspects. This works well in preventing BEVDet from over-fitting.

In addition, we upgrade the classical Non-Maximum Suppression (NMS) strategy for improving its adaptability in the 3D object detection scenario. The inference process is further sped up by removing the sequentially executed operators. With these modifications, BEVDet offers an outstanding trade-off between accuracy and inference latency among existing paradigms. On nuScenes [1] `val` set, the high-speed version, BEVDet-Tiny, achieves superior accuracy (*i.e.*, 31.2% mAP and 39.2% NDS) with an image size of 704×256, which is merely 1/8 of the competitors' (*i.e.*, 29.5% mAP and 37.2% NDS with 1600×900 image size in FCOS3D [49]). Scaling down the image size reduces the computational budget by 89% and offers a dramatic acceleration of 9.2 times (*i.e.*, BEVDet with 215.3 GFLOPs and 15.6 FPS *v.s.* FCOS3D with 2,008.2 GFLOPs and 1.7 FPS). By constructing another high-precision configuration dubbed BEVDet-Base, we report a new record of 39.3% mAP and 47.2% NDS. Moreover, compared to the existing paradigms, explicitly encoding features in BEV space makes BEVDet talented at perceiving the targets' translation, scale, orientation, and velocity. More characteristic of BEVDet can be found in ablation study.

2   Related Works

2.1   Vision-based 2D Perception

Image Classification The renaissance of deep learning for vision-based 2D perception can be dated back to AlexNet [19] for image classification. From then on, the research community keeps pushing the performance boundary of image encoder by giving raise to residual [14], high-resolution [43], attention-based [7], and many other types of structures [8,15,16,35,45]. And at the same time, the powerful image encoding capacity also boosts the performance of other complicated tasks like object detection [23,38], semantic segmentation [18,52], human pose estimation [17,43], and so on. As a simple task, the solution pattern of image classification is dominated by Softmax [19] and its derivatives. Determined

by the network structure, the capacity of the image encoders plays a vital role in this problem and is the main concern in the research community.

Object Detection Common object detection, demanding both category labels and the locating bounding boxes of all pre-defined objects, is a more complicated task where paradigms also play a vital role. Two-stage method Faster R-CNN [38], one-stage method RetinaNet [23], and their derivatives [2, 4, 13, 46, 57] are the dominant methods in this area [24, 41]. Inspired by Mask R-CNN [13], multi-task learning has been an appealing paradigm in both the research and industry community, owing to its great potential for saving computational resources by sharing backbone and promoting tasks by training jointly. The great impact of paradigm innovation in this area inspires us to exploit superior paradigms for better perception performance in the scene of autonomous driving, where the tasks are even more complicated and multi-task learning is rather appealing.

2.2   Semantic Segmentation in BEV

One of the main perception tasks in autonomous driving is to vectorially restore the map of its surrounding environment. This can be achieved by semantic segmentation in BEV for the targets like drivable areas, car parking, lane dividers, stopping lines, and so on. The vision-based methods with leading performance in benchmark [1] are always with a similar framework [31, 33, 39, 54]. In this framework, there are four main components: an image-view encoder for encoding features in image view, a view transformer for transforming the features from image view to BEV, a BEV encoder for further encoding the feature in BEV, and a head for pixel-wise classification. The success of this pipeline in BEV semantic segmentation encourages us to extend it to the 3D object detection task, expecting that the features in BEV can work well in capturing some targets of 3D objects like scale, orientation, and velocity. Besides, we are also seeking a scalable paradigm in which multi-tasks learning can be achieved with both high accuracy and high efficiency.

2.3   Vision-based 3D Object Detection

3D object detection is another pivotal perception task in autonomous driving. In the last few years, KITTI [9] benchmark has fueled the rapid development of monocular 3D object detection [20, 27, 29, 36, 47, 48, 58, 61, 64]. However, the limited data and the single view make it incapable of developing more complicated tasks. Recently, some large-scale benchmark [1, 44] have been released with more data and multiple views, offering new perspectives toward the paradigm development in multi-camera 3D object detection. Based on these benchmarks, some multi-camera 3D object detection paradigms have been developed with competitive performance. For example, inspired by the success of FCOS [46] in 2D detection, FCOS3D [49] treats the 3D object detection problem as a 2D object detection problem and conducts perception just in image view. Benefitting from

the strong spatial correlation of the targets' attribute with the image appearance, it works well in predicting this but is relatively poor in perceiving the targets' translation, velocity, and orientation. Following DETR [3], DETR3D [51] proposes to detect 3D objects in an attention pattern, which has similar accuracy as FCOS3D. Although DETR3D requires just half the computational budget, the complex calculation pipeline slows down its inference speed to the same level as FCOS3D. PGD [50] further develops the FCOS3D paradigm by searching and resolving with the outstanding shortcoming (*i.e.* the prediction of the targets' depth). This offers a remarkable accuracy improvement on the baseline but at the cost of more computational budget and additional inference latency. The existing paradigms have limited tradeoffs between accuracy and time-efficiency. This motivates us to seek and develop new ones for substantially pushing the performance boundary in this area.

There are some pioneers [5, 21, 36], who have exploited the 3D object detection task in BEV. Among them, also inspired by Lift-Splat-Shoot [33], [36] is the most similar one as ours. They apply the Lift-Splat-Shoot paradigm on monocular 3D object detection and make it competitive in the KITTI [9] benchmark by referring to LiDARs for the supervision on depth prediction. A close idea can be found in the concurrent work of DD3D [32]. Differently, without the reliance on LiDARs, we upgrade this paradigm by constructing an exclusive data augmentation strategy based on the decoupling effect of the view transformer. This is a more feasible way and plays the essential role in enabling the BEVDet paradigm to perform competitively among existing methods.

3   Methodology

3.1   Network Structure

As illustrated in Fig. 1, BEVDet with a modular design consists of four kinds of modules: an image-view encoder, a view transformer, a BEV encoder, and a task-specific head. We study the feasibility of BEVDet by constructing several derivatives with different structures as listed in Tab. 1.

Image-view Encoder The image-view encoder encodes the input images into high-level features. To exploit the power of multi-resolution features, the image-view encoder includes a backbone for high-level feature extraction and a neck for multi-resolution feature fusion. By default, we use the classical ResNet [14] and the up-to-date attention-based SwinTransformer [26] as backbone for prototype study. The substitutions include DenseNet [16], HRNet [43] and so on. With respect to the neck module, we use the classical FPN [22] and the neck structure proposed in [33], which is named FPN-LSS in the following. FPN-LSS simply upsamples the feature with 1/32 input resolution to 1/16 input resolution and concatenates it with the one generated by the backbone. More complicated neck modules have not been exploited like PAFPN [25], NAS-FPN [10] and so on.

Table 1. The components of BEVDet. '-number' denotes the number of channels in this module. Lift-Splat-Shoot-64-0.4×0.4 denotes the view transformer proposed in [33]. The output feature has a channel number of 64 and a resolution of 0.4 meters.

Module	BEVDet-Base	BEVDet-Tiny	BEVDet-R50	BEVDet-R101
Input Resolution	1600×640	704×256	704×256	704×256
Image-view Encoder	SwinTransformer-Base	SwinTransformer-Tiny	ResNet-50	ResNet-101
	FPN-LSS-512	FPN-LSS-512	FPN-512	FPN-256
View Transformer	Lift-Splat-Shoot-64 -0.4×0.4	Lift-Splat-Shoot-64 -0.8×0.8	Lift-Splat-Shoot-80 -0.8×0.8	Lift-Splat-Shoot-64 -0.8×0.8
BEV Encoder	2×Basic-128	2×Basic-128	2× Basic-160	1× Basic-128
	2×Basic-256	2×Basic-256	2× Basic-320	1× Basic-256
	2×Basic-512	2×Basic-512	2× Basic-640	1× Basic-512
	FPN-LSS-512	FPN-LSS-256	FPN-LSS-256	FPN-LSS-128
Head	CenterPoint Head [56]			

View Transformer The view transformer transforms the feature from image view to BEV. We apply the view transformer proposed in [33] to construct the BEVDet prototype. The adopted view transformer takes the image-view feature as input and densely predicts the depth through a classification manner. Then, the classification scores and the derived image-view feature are used in rendering the predefined point cloud. Finally, the BEV feature can be generated by applying a pooling operation along the vertical direction (*i.e.*, Z coordinate axis as illustrated in Fig. 1). In practice, we extend the default range of depth prediction to $[1, 60]$ meters with an interval of $1.25 \times r$, where r denotes the resolution of the output features.

BEV Encoder The BEV encoder further encodes the feature in the BEV space. Though the structure is similar to that of the image-view encoder with a backbone and a neck, it perceives some pivotal cues with high precision like scale, orientation, and velocity, as they are defined in the BEV space. We follow [33] to utilize ResNet [14] with classical residual block to construct the backbone and combine the features with different resolutions by applying FPN-LSS.

Head The task-specific head is constructed upon the BEV feature. In common sense [1], 3D object detection in automatic pilot aims at the position, scale, orientation, and speed of movable objects like pedestrians, vehicles, barriers, and so on. Without any modification, we directly adopt the 3D object detection head in the first stage of CenterPoint [56] for prototype verification and fair comparison with the LiDAR-based pipelines like PointPillar [21] and VoxelNet [60]. The second refinement stage of CenterPoint has not been applied.

3.2   The Customized Data Augmentation Strategy

The Isolated View Spaces The view transformer [33] transforms the feature from image view to BEV in a pixel-wise manner. Specifically, given a pixel in

the image plane $\mathbf{p}_{image} = [x_i, y_i, 1]^T$ with a specific depth d, the corresponding coordinate in the 3D space is:

$$\mathbf{p}_{camera} = \mathbf{I}^{-1}(\mathbf{p}_{image} * d) \tag{1}$$

where $\mathbf{I}$ is the 3×3 camera intrinsic matrix. Common data augmentation strategies with operations like flipping, cropping, and rotating can be formulated as a 3×3 transformation matrix $\mathbf{A}$ [17]. When a data augmentation strategy is applied on the input image (*i.e.*, $\mathbf{p}'_{image} = \mathbf{A}\mathbf{p}_{image}$), inverse transformation $\mathbf{A}^{-1}$ should be applied in the view transformation [33] to maintain the spatial consistency between the features and the targets in the BEV space:

$$\mathbf{p}'_{camera} = \mathbf{I}^{-1}(\mathbf{A}^{-1}\mathbf{p}'_{image} * d) = \mathbf{p}_{camera} \tag{2}$$

According to Eq. 2, the augmentation strategy applied in image view space will not change the spatial distribution of the features in BEV space. This makes performing complicated data augmentation strategies in the image view space feasible for BEVDet.

BEV Space Learning with Data Augmentation. With respect to the learning in the BEV space, the number of data is less than that in the image view space as each sample contains multiple camera images (*e.g.* each sample in the nuScenses benchmark contains 6 images [1]). The learning in the BEV space is thus prone to fall into over-fitting. As the view transformer isolates the two view spaces in the augmentation perspective, we construct another augmentation strategy specific for the regularization effect on the learning in BEV space. Following the up-to-date LiDAR-based methods [34, 55, 56, 63], common data augmentation operations in 2D space are adopted including flipping, scaling, and rotating. In practice, the operations are conducted both on the output feature of the view transformer and the 3D object detection targets to keep their spatial consistency. It is worth noting that this data augmentation strategy is built upon the precondition that the view transformer can decouple the image-view encoder from the subsequent module. This is a specific characteristic of BEVDet and may not be effective in the other methods [49–51].

3.3   Scale-NMS

The spatial distribution of different categories in the BEV space is quite different from that in the image-view space. In the image-view space, all categories share a similar spatial distribution due to the perspective imaging mechanism of the cameras. Therefore, a classical Non-Maximum Suppression (NMS) [40] strategy with a fixed threshold can work well in adjusting the predicted results of all categories to agree with the priors (*e.g.*, the bounding box Intersection-Over-Union (IOU) indicator between two instances is always below a specific threshold of 0.5 in 2D object detection [11, 23, 37, 38, 46]). However, it is different in the BEV space. In the BEV space, the occupied areas of various classes are
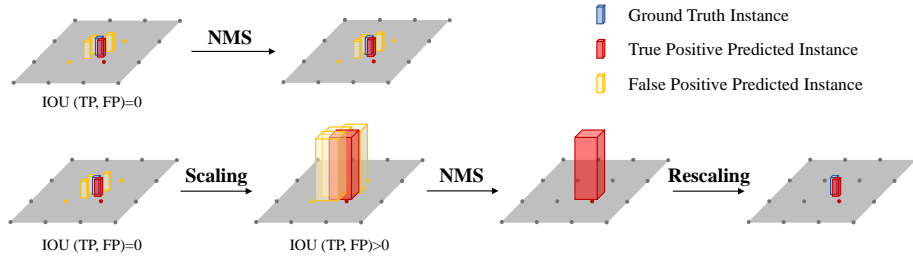
Fig. 2. Illustration of the comparison between classical NMS and the proposed Scale-NMS.

intrinsically different and the overlap between instances should be closed to zero. As a result, the distribution of IOU between predicted results varies by category. For example, as illustrated in Fig. 2, objects like pedestrians and traffic cones occupy a small area in the ground plane, which is always smaller than the output resolution of the algorithm (*e.g.*, 0.8 meters in CenterPoint [56]). Common object detection paradigms [23, 38, 46, 56] redundantly generate predictions. The small occupied area of each object may make the redundant results to have no intersection with the true positive one. This deactivates the classical NMS who rely on IOU to access the spatial relationship between the true positives and the false positives.

To overcome the aforementioned problems, we proposed Scale-NMS in this paper. Scale-NMS scales the size of each object according to its category before performing the classical NMS algorithm. In this way, the distribution of IOU between the true positives and the redundant results is modulated to match up with the classical NMS. As illustrated in the second row of Fig. 2, in predicting small objects, Scale-NMS builds the spatial relationship between results by scaling up the object size, which enables the classical NMS to drop the redundant ones according to the IOU indicator. In practice, we apply Scale-NMS to all categories except for the barrier as its size is various. The scaling factors are category-specific. They are generated by hyper-parameter searching on the validation set.

4 Experiment

4.1 Experimental Settings

Dataset We conduct comprehensive experiments on the large-scale benchmark nuScenes [1]. The nuScenes benchmark includes 1000 scenes with images from 6 cameras. It is the up-to-date popular benchmark for vision-based 3D object detection [32, 49–51] and BEV semantic segmentation [31, 33, 39, 54]. The scenes are officially split into 700/150/150 scenes for training/validation/testing. There are up to 1.4M annotated 3D bounding boxes for 10 classes: car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, barrier, and traffic

cone. Following CenterPoint [56], we define the region of interest (ROI) within 51.2 meters in the ground plane with a resolution (*i.e.*, the size of voxel in CenterPoint [56]) of 0.8 meters by default.

Evaluation Metrics For 3D object detection, we report the official predefined metrics: mean Average Precision (mAP), Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE), Average Attribute Error (AAE), and NuScenes Detection Score (NDS). The mAP is analogous to that in 2D object detection [24] for measuring the precision and recall, but defined based on the match by 2D center distance on the ground plane instead of the Intersection over Union (IOU) [1]. NDS is the composite of the other indicators for comprehensively judging the detection capacity. The remaining metrics are designed for calculating the positive results' precision on the corresponding aspects (*e.g.*, translation, scale, orientation, velocity, and attribute).

Training Parameters Models are trained with AdamW [28] optimizer, in which gradient clip is exploited with learning rate 2e-4, a total batch size of 64 on 8 NVIDIA GeForce RTX 3090 GPUs. For ResNet [14] based image-view encoder, we apply a step learning rate policy, which drops the learning rate at epoch 17 and 20 by a factor of 0.1. With respect to SwinTransformer [26] based image-view encoder, we apply a cyclic policy [53], which linearly increases the learning rate from 2e-4 to 1e-3 in the first 40% schedule and linearly decreases the learning rate from 1e-3 to 0 in the remainder epochs. By default, the total schedule is terminated within 20 epochs.

Data Processing We use $W_{in} \times H_{in}$ to denote the width and height of the input image. By default in the training process, the source images with 1600×900 resolution [1] are processed by random flipping, random scaling with a range of $s \in [W_{in}/1600 - 0.06, W_{in}/1600 + 0.11]$, random rotating with a range of $r \in [-5.4°, 5.4°]$, and finally cropping to a size of $W_{in} \times H_{in}$. The cropping is conducted randomly in the horizon direction but is fixed in the vertical direction (*i.e.*, $(y_1, y_2) = (max(0, s * 900 - H_{in}), y_1 + H_{in})$, where y_1 and y_2 are the upper bound and the lower bound of the target region.) In the BEV space, the input feature and 3D object detection targets are augmented by random flipping, random rotating with a range of $[-22.5°, 22.5°]$, and random scaling with a range of $[0.95, 1.05]$. Following CenterPoint [56], all models are trained with CBGS [62]. In testing time, the input image is scaled by a factor of $s = W_{in}/1600 + 0.04$ and cropped to $W_{in} \times H_{in}$ resolution with a region defined as $(x_1, x_2, y_1, y_2) = (0.5 * (s * 1600 - W_{in}), x_1 + W_{in}, s * 900 - H_{in}, y_1 + H_{in})$.

Inference Speed We conduct all experiments based on MMDetection3D [6]. All inference speeds and computational budgets are tested without data augmentation. For monocular paradigms like FCOS3D [49] and PGD [50], the inference

Table 2. Comparison of different paradigms on the nuScenes `val` set. † initialized from a FCOS3D backbone. § with test-time augmentation. # with model ensemble.

Methods	Image Size	#param.	GFLOPs	Modality	**mAP**↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	**NDS**↑	FPS
VoxelNet [56]	-	-	-	LiDAR	0.564	-	-	-	-	-	0.648	-
PointPillar [56]	-	-	-	LiDAR	0.503	-	-	-	-	-	0.602	-
CenterNet [59]	-	-	-	Camera	0.306	0.716	0.264	0.609	1.426	0.658	0.328	-
FCOS3D [49]	1600×900	52.5M	2,008.2	Camera	0.295	0.806	0.268	0.511	1.315	**0.170**	0.372	1.7
DETR3D [51]	1600×900	51.3M	1,016.8	Camera	0.303	0.860	0.278	0.437	0.967	0.235	0.374	2.0
PGD [50]	1600×900	53.6M	2,223.0	Camera	0.335	0.732	0.263	0.423	1.285	0.172	0.409	1.4
BEVDet-Tiny	704×256	53.7M	215.3	Camera	0.312	0.691	0.272	0.523	0.909	0.247	0.392	**15.6**
BEVDet-Base	1600×640	126.6M	2,962.6	Camera	**0.393**	0.608	0.259	0.366	0.822	0.191	**0.472**	1.9
FCOS3D†§# [49]	1600×900	-	-	Camera	0.343	0.725	0.263	0.422	1.292	**0.153**	0.415	-
DETR3D† [51]	1600×900	51.3M	-	Camera	0.349	0.716	0.268	0.379	0.842	0.200	0.434	-
PGD†§ [50]	1600×900	53.6M	-	Camera	0.369	0.683	0.260	0.439	1.268	0.185	0.428	-
BEVDet-Base§	1600×640	126.6M	-	Camera	**0.397**	**0.595**	**0.257**	**0.355**	**0.818**	0.188	**0.477**	-

speeds are divided by a factor of 6 (*i.e.* the number of images in a single sample [1]), as they take each image as an independent sample. It is worth noting that, the dividing operation may not be the optimal method, as processing in the batch pattern can speed up the inference of monocular paradigms. We accelerate the proposed BEVDet paradigm by replacing the accumulative sum operation in the view transformation with another equivalent implementation. Details can be found in the ablation study section.

4.2   Benchmark Results

nuScenes `val` set We comprehensively compare the proposed BEVDet with other paradigms like FCOS3D [49], its upgraded version PGD [50], and DETR3D [51]. Their numbers of parameters, computational budget, inference speed, and accuracy on the nuScenes `val` set are all listed in Tab. 2.

As a high-speed version dubbed BEVDet-Tiny, we set the number of parameters close to competitors and equip it with a small input resolution of 704×256. With merely 1/8 input size of the competitors (*i.e.*, 704×256 for BEVDet-Tiny *v.s.* 1600×900 for FCOS3D, DETR3D, and PGD), BEVDet-Tiny requires just 215.3 GFLOPs computational budget and can be processed in 15.6 FPS. It scores 31.2% mAP and 39.2% NDS, which has a superior accuracy than FCOS3D (29.5% mAP and 37.2% NDS) and DETR3D (30.3% mAP and 37.4% NDS). However, it requires far less computational budget (2,008.2 GFLOPs of FCOS3D, 1,016.8 GFLOPs of DETR3D) and has a faster inference speed (1.7 FPS of FCOS3D, 2.0 FPS of DETR3D). BEVDet-Base with 1600×640 input resolution requires 2962.6 GFLOPs scores 39.3% mAP and 47.2% NDS. With a competitive inference speed, BEVDet-Base outperforms all published results. It significantly exceeds the previous leading method PGD by a margin of +5.8% mAP and +6.3% NDS. It is worth noting that though the computational budget of BEVDet-Base is nearly 3 times that of DETR3D [51], BEVDet-Base can be process at a comparable speed of 1.9 FPS. The straightforward design enables BEVDet to run faster than the existing paradigms.

Considering the translation (ATE), scale (ASE), orientation (AOE), velocity (AVE), and attribute (AAE) error of the truly positive results, BEVDet works

Table 3. Comparison with the state-of-the-art methods on the nuScenes `test` set. †
pre-train on DDAD [12].

Methods	Modality	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
PointPillars (Light) [21]	LiDAR	0.305	0.517	0.290	0.500	0.316	0.368	0.453
CenterFusion [30]	Camera & Radar	0.326	0.631	0.261	0.516	0.614	0.115	0.449
CenterPoint [56]	Camera & LiDAR & Radar	0.671	0.249	0.236	0.350	0.250	0.136	0.714
MonoDIS [42]	Camera	0.304	0.738	0.263	0.546	1.553	0.134	0.384
CenterNet [59]	Camera	0.338	0.658	0.255	0.629	1.629	0.142	0.400
FCOS3D [49]	Camera	0.358	0.690	0.249	0.452	1.434	**0.124**	0.428
PGD [50]	Camera	0.386	0.626	0.245	0.451	1.509	0.127	0.448
BEVDet	Camera	**0.422**	**0.529**	**0.236**	**0.395**	**0.979**	0.152	**0.482**
DD3D† [32]	Camera	0.418	0.572	0.249	0.368	1.014	0.124	0.477
DETR3D† [51]	Camera	0.386	0.626	0.245	0.394	0.845	0.133	0.479

Table 4. Ablation study for the data augmentation strategy on the nuScenes `val` set.
IDA denotes Image-view-space Data Augmentation. BDA denotes BEV-space Data
Augmentation. BE denotes BEV Encoder.

ID	IDA	BDA	BE	mAP-best↑	NDS-best↑	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
A			✓	**0.230** (e4)	**0.310** (e14)	0.174 (-5.6%)	0.283 (-2.7%)	0.734	0.343	0.664	1.262	0.298
B	✓		✓	**0.205** (e10)	**0.308** (e14)	0.178 (-2.7%)	0.303 (-0.5%)	0.738	0.288	0.615	1.030	0.217
C		✓	✓	**0.262** (e11)	**0.357** (e14)	0.236 (-2.6%)	0.348 (-0.9%)	0.717	0.274	**0.514**	0.976	0.221
D	✓	✓	✓	**0.316** (e17)	**0.393** (e19)	0.312 (**-0.4%**)	0.392 (**-0.1%**)	**0.691**	**0.272**	0.523	**0.909**	0.247
E				**0.231** (e10)	**0.307** (e10)	0.215 (-1.6%)	0.306 (-0.1%)	0.777	0.283	0.703	1.111	0.249
F	✓			**0.276** (e14)	**0.347** (e17)	0.269 (-0.7%)	0.345 (-0.2%)	0.734	0.274	0.673	0.994	0.217
G		✓		**0.253** (e12)	**0.345** (e15)	0.224 (-2.9%)	0.337 (-0.8%)	0.734	0.281	0.543	0.983	**0.211**
H	✓	✓		**0.299** (e20)	**0.373** (e20)	0.299 (-0.0%)	0.373 (-0.0%)	0.726	0.273	0.536	0.950	0.278

well in predicting the targets' translation, scale, orientation, and velocity, which
is consistent with common sense that it is easier for an agent to capture these
kinds of information in BEV. However, BEVDet performs poorly in predicting
the targets' attributes when compared with the image-view-based methods like
FCOS3D and PGD. We conjecture that the attribute judgment relies on the
appearance cues, which is easier for agents to perceive in the image view. The
combination of these two views is a promising solution to this problem, which
will be studied in future work.

nuScenes `test` set For the nuScenes `test` set, we train BEVDet-Base config-
uration on the `train` and `val` sets. Single model with test time augmentation is
adopted. As listed in Tab. 3, BEVDet ranks first on the nuScenes vision-based
3D objection leaderboard with scores of 42.2% mAP and 48.2% NDS, surpass-
ing the previous leading method PGD [50] by +3.6% mAP and +3.4% NDS.
This has been comparable with those relied on LiDAR sensor for pre-training
like DD3D [32] and DETR3D [51]. It is also worth noting that the accuracy
of vision-based BEVDet is comparable with the classical LiDAR-based method
PointPillars [21] (*i.e.* 30.5% mAP and 45.3% NDS).

4.3   Ablation Studies

Data Augmentation With BEVDet-Tiny in Tab. 1, we study how the perfor-
mance of BEVDet is developed by the customized data augmentation strategy.

We adopt a fixed training schedule of 20 epochs and report both the best performances during the training process and the final performances at the last epoch. By reporting and comparing the two, we analyze how the data augmentation strategy affects the performances at the saturation point and to what degree does the data augmentation alleviates the over-fitting problem. Some key factors are considered including Image-view-space Data Augmentation (IDA), BEV-space Data Augmentation (BDA), and BEV Encoder (BE). We listed the performance of different configurations in Tab. 4.

As a baseline, we simply replace the head's input feature in LiDAR-based method CenterPoint [56] with the one generated by the view transformer proposed in [33]. In this configuration Tab. 4 (A), all augmentation strategies are absent. During the training process, indicator mAP becomes saturated early at epoch 4 with 23.0% and falls into over-fitting in the following epochs. Finally, the performance at epoch 20 is merely 17.4% with a drop from the best by -5.6%, which is far poor than the image-view-based method FCOS3D (29.5%).

By applying Image-view-space Data Augmentation (IDA) in configuration Tab. 4 (B), the saturation of the training process is postponed to epoch 10 (20.5%) and finally scores 17.8%. The best performance of this configuration is even worse than the baseline (*i.e.*, Tab. 4 (A)). In contrast, configuration Tab. 4 (C) with BEV-space Data Augmentation (BDA) peaks at epoch 15 with 26.2% mAP and finally scores 23.6% mAP at epoch 20. This surpasses the baseline by a large margin of +3.2% mAP at the peak point. BDA plays a more important role than IDA in training BEVDet. By combining both IDA and BDA in configuration Tab. 4 (D), the mAP performance peaks at epoch 17 with 31.6% and finally scores 31.2% at epoch 20. Compared with the baseline, the combined data augmentation strategy offers a significant performance boost of +8.6% mAP at the peak point. The performance degeneration at epoch 20 is reduced to -0.4%. It is worth noting that, IDA has a negative impact on the performance when BDA is absent but has a positive impact on the contrary.

To study the impact of BEV Encoder (BE), we remove BE in configuration Tab. 4 (E, F, G, and H). Comparing configuration Tab. 4 (D) with (H), BE improves the BEVDet's accuracy by +1.7% mAP, indicating that it is one of the key components in constructing the performance of BEVDet. By comparing configuration Tab. 4 (F) with (G), we found that IDA can offer a positive impact when BDA is absent, which is opposite when BE is present. We conjecture that the strong perception capacity of BEV Encoder can only be built upon the presence of BDA. This can be verified by comparing the best performance of configuration Tab. 4 (A, B, C, and D) with (E, F, G, and H) respectively.

Scale-NMS We adopt BEVDet-Tiny in Tab. 1 for ablation study on the NMS strategy. As shown in Tab. 5, category by category, we compare Scale-NMS with the classical NMS and the Circular-NMS proposed in CenterPoint [56]. The proposed Scale-NMS significantly boosts the performance on the categories with a small occupied area like pedestrians (+4.8% AP) and traffic cones(+7.5% AP). The other categories with relatively large sizes also benefit from Scale-NMS

Table 5. Ablation study for the NMS strategy on the nuScenes `val` set.

Methods	mAP	Car	Truck	Bus	Trailer	C-Vehicle	Pedestrian	Motorcycle	Bicycle	Traffic Cone	Barrier
NMS	0.295	0.512	0.220	0.305	0.153	0.069	0.297	**0.273**	**0.225**	0.425	0.467
Circular-NMS	0.298	**0.516**	0.210	0.308	0.149	0.066	0.295	0.272	0.212	0.451	**0.498**
Scale-NMS	**0.312**	0.512	**0.223**	**0.313**	**0.160**	**0.072**	**0.345**	**0.273**	**0.225**	**0.500**	**0.498**

Table 6. Ablation study for the resolutions of BEVDet on the nuScenes `val` set.

ID	Input Resolution	BEV Resolution	mAP	NDS	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	GFLOPs	FPS
A	704×256	0.8 Meter	0.312	0.392	0.691	0.272	0.523	0.909	0.247	215.3	15.6
B	1056×384	0.8 Meter	0.333	0.410	0.661	0.265	0.509	0.886	0.243	370.5	8.9
C	704×256	0.4 Meter	0.315	0.410	0.653	0.274	0.492	0.851	0.254	438.4	10.0
D	1056×384	0.4 Meter	0.348	0.417	0.644	0.266	0.475	0.916	0.264	593.6	7.1
E	1408×512	0.4 Meter	0.360	0.438	0.638	0.266	0.427	0.878	0.213	824.6	5.0

like buses (+0.8% AP), trucks (+0.3% AP), trailers (+0.7% AP), and construct vehicles (+0.3% AP). The overall performance mAP is thus boosted from 29.5% to 31.2% with an improvement of +1.7%.

Resolution The resolution of the signal channel is vital for BEVDet. It not only affects the accuracy of the models but also plays a key role in the computational budget and inference latency. As BEVDet involves two view spaces, two main channel resolutions are studied here: the resolution of the input image and the resolution of the BEV encoder's input features. We perform several ablation experiment in Tab. 6 with some typical settings. According to the results, the resolution of input image has a large impact on the accuracy. For example, BEVDet with 1408×512 input size Tab. 6 (E) has a +4.5% mAP superiority on that with 704×256 input size Tab. 6 (C). It is worth noting that with the increasing of input size, the increment of the BEVDet computational budget is limited as the computational budget of the BEV encoder and heads is unchanged. Besides, a larger input size also has a consistently positive impact on predicting the targets' translation, scale, and orientation.

With respect to the resolution of the BEV encoder's input features, it can also be regarded as the voxel size in most classical LiDAR based methods [56, 60]. Improving the resolution of the BEV encoder's input features can boost the accuracy of models on mAP, mATE, and mAOE indicators, but at the cost of higher computational budget and inference latency.

Backbone Type in the Image-view Encoder We study the effect of backbone type in the image-view encoder by constructing 3 derivatives of BEVDet with different structures in Tab. 1. They are all constructed under the principle of containing a similar amount of parameters. As listed in Tab. 7, two input resolutions are adopted. When changing the backbone type of the image-view encoder from ResNet-R50 [14] into SwinTransformer-Tiny [26] with a low input resolution of 704×256, the gains are +1.4% mAP and +1.3% NDS (*i.e.*, BEVDet-R50 with 29.8% mAP and 37.9% NDS *v.s.* BEVDet-Tiny with 31.2% mAP and 39.2% NDS). BEVDet-R50 is particularly stronger in predicting the

Table 7. Ablation study for the image-view encoder on the nuScenes `val` set.

Configuration	Input Resolution	mAP	NDS	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	#param.	GFLOPs	FPS
BEVDet-R50	704×256	0.298	0.379	0.725	0.279	0.589	0.860	0.245	53.3M	183.8	16.7
BEVDet-R101	704×256	0.302	0.381	0.722	0.269	0.543	0.900	0.269	54.1M	223.6	14.3
BEVDet-Tiny	704×256	0.312	0.392	0.691	0.272	0.523	0.909	0.247	53.7M	215.3	15.6
BEVDet-R50	1056×384	0.318	0.389	0.718	0.272	0.553	0.897	0.258	53.3M	311.8	11.4
BEVDet-R101	1056×384	0.330	0.396	0.702	0.272	0.534	0.932	0.251	54.1M	452.0	9.3
BEVDet-Tiny	1056×384	0.333	0.410	0.661	0.265	0.509	0.886	0.243	53.7M	370.5	8.9
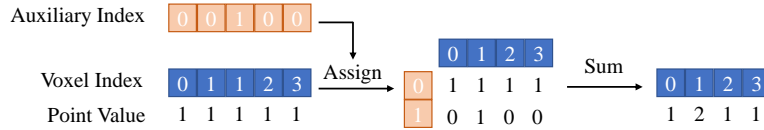
Fig. 3. Combining the features with the auxiliary indexes.

targets' velocity, while BEVDet-Tiny has superior performance in predicting the target's translation and orientation. With respect to BEVDet-R101, the gains are merely +0.4% mAP and +0.2% NDS on BEVDet-R50 when a small input size of 704×256 is adopted. However, the gains are +1.2% mAP and +0.7% NDS when a larger input size of 1056×384 is applied. We conjecture that a larger receptive field plays an important role in scaling up the input size.

Acceleration The adopted view transformation [33] conducts accumulate sum in combining the features within the same voxel. However, the inference latency of this operation is proportional to the overall point number. To remove this operation, as illustrated in Fig. 3, we introduce an auxiliary index to record how many times does the same voxel index has been present previously. With this auxiliary index and the voxel index, we assign the points into a 2-D matrix and combine the features within the same voxel with a sum operation alone the auxiliary axis. Under the pre-condition that the camera intrinsic and extrinsic parameters are fixed in the inference time, the auxiliary index and the voxel index are fixed and can be calculated in the initialization phase [33]. With this modification, we reduce the inference latency of BEVDet-Tiny by 53.3% (*i.e.*, from 137 milliseconds to merely 64 milliseconds). It is worth noting that, this modification requires extra memory which is determined by the number of voxels and the maximum value of the auxiliary index. In practice, we limit the maximum value of the auxiliary index to 300 and drop the remaining points. This operation has negligible impact on the model accuracy.

5   Conclusion

In this paper, we propose BEVDet, a powerful and scalable paradigm for multi-camera 3D object detection. BEVDet is constructed by referring to the success of

solving semantic segmentation in BEV and is developed mainly by constructing an exclusive data augmentation strategy. In the large-scale benchmark nuSenses, BEVDet significantly pushes the performance boundary and is particularly good at predicting the targets' translation, scale, orientation, and velocity. Future works will focus on (1) improving the performance of BEVDet, particularly on targets' attribute prediction. (2) studying multi-task learning based on BEVDet.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: High Quality Object Detection and Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: Proceedings of the European Conference on Computer Vision. pp. 213–229. Springer (2020)
4. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al.: Hybrid Task Cascade for Instance Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4974–4983 (2019)
5. Chen, Y., Liu, S., Shen, X., Jia, J.: DSGN: Deep Stereo Geometry Network for 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12536–12545 (2020)
6. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d (2020)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. In: Proceedings of the International Conference on Learning Representations (2020)
8. Gao, S., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.H.: Res2Net: A New Multi-scale Backbone Architecture. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2012)
10. Ghiasi, G., Lin, T.Y., Le, Q.V.: NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7036–7045 (2019)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
12. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3D Packing for Self-Supervised Monocular Depth Estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2485–2494 (2020)

13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
15. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861 (2017)
16. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017)
17. Huang, J., Zhu, Z., Guo, F., Huang, G.: The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5700–5709 (2020)
18. Kirillov, A., Wu, Y., He, K., Girshick, R.: PointRend: Image Segmentation as Rendering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9799–9808 (2020)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems **25**, 1097–1105 (2012)
20. Kumar, A., Brazil, G., Liu, X.: GrooMeD-NMS: Grouped Mathematically Differentiable NMS for Monocular 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8973–8983 (2021)
21. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast Encoders for Object Detection from Point Clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125 (2017)
23. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: Proceedings of the International Conference on Computer Vision. pp. 2980–2988 (2017)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Proceedings of the European Conference on Computer Vision. pp. 740–755. Springer (2014)
25. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path Aggregation Network for Instance Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8759–8768 (2018)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: Proceedings of the International Conference on Computer Vision. pp. 10012–10022 (2021)
27. Liu, Z., Zhou, D., Lu, F., Fang, J., Zhang, L.: AutoShape: Real-Time Shape-Aware Monocular 3D Object Detection. In: Proceedings of the International Conference on Computer Vision. pp. 15641–15650 (2021)
28. Loshchilov, I., Hutter, F.: DECOUPLED WEIGHT DECAY REGULARIZATION. In: Proceedings of the International Conference on Learning Representations (2019)
29. Lu, Y., Ma, X., Yang, L., Zhang, T., Liu, Y., Chu, Q., Yan, J., Ouyang, W.: Geometry Uncertainty Projection Network for Monocular 3D Object Detection.

In: Proceedings of the International Conference on Computer Vision. pp. 3111–3121 (2021)

30. Nabati, R., Qi, H.: CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1527–1536 (2021)

31. Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B.: Cross-View Semantic Segmentation for Sensing Surroundings. IEEE Robotics and Automation Letters **5**(3), 4867–4873 (2020)

32. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is Pseudo-Lidar needed for Monocular 3D Object detection? In: Proceedings of the International Conference on Computer Vision. pp. 3142–3152 (2021)

33. Philion, J., Fidler, S.: Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In: Proceedings of the European Conference on Computer Vision. pp. 194–210. Springer (2020)

34. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 5105–5114 (2017)

35. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing Network Design Spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10428–10436 (2020)

36. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical Depth Distribution Network for Monocular 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021)

37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)

38. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems **28**, 91–99 (2015)

39. Roddick, T., Cipolla, R.: Predicting Semantic Map Representations from Images using Pyramid Occupancy Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11138–11147 (2020)

40. Rosenfeld, A., Thurston, M.: Edge and Curve Detection for Visual Scene Analysis. IEEE Transactions on computers **100**(5), 562–569 (1971)

41. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A Large-scale, High-quality Dataset for Object Detection. In: Proceedings of the International Conference on Computer Vision. pp. 8430–8439 (2019)

42. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling Monocular 3D Object Detection. In: Proceedings of the International Conference on Computer Vision. pp. 1991–1999 (2019)

43. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep High-Resolution Representation Learning for Human Pose Estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5693–5703 (2019)

44. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020)

45. Tan, M., Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Proceedings of the International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)

46. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully Convolutional One-Stage Object Detection. In: Proceedings of the International Conference on Computer Vision. pp. 9627–9636 (2019)
47. Wang, L., Du, L., Ye, X., Fu, Y., Guo, G., Xue, X., Feng, J., Zhang, L.: Depth-conditioned Dynamic Message Propagation for Monocular 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 454–463 (2021)
48. Wang, L., Zhang, L., Zhu, Y., Zhang, Z., He, T., Li, M., Xue, X.: Progressive Coordinate Transforms for Monocular 3D Object Detection. In: Advances in Neural Information Processing Systems (2021)
49. Wang, T., Zhu, X., Pang, J., Lin, D.: FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection. arXiv preprint arXiv:2104.10956 (2021)
50. Wang, T., Zhu, X., Pang, J., Lin, D.: Probabilistic and Geometric Depth: Detecting Objects in Perspective. arXiv preprint arXiv:2107.14160 (2021)
51. Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries. arXiv preprint arXiv:2110.06922 (2021)
52. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified Perceptual Parsing for Scene Understanding. In: Proceedings of the European Conference on Computer Vision. pp. 418–434 (2018)
53. Yan, Y., Mao, Y., Li, B.: SECOND: Sparsely Embedded Convolutional Detection. Sensors **18**(10), 3337 (2018)
54. Yang, W., Li, Q., Liu, W., Yu, Y., Ma, Y., He, S., Pan, J.: Projecting Your View Attentively: Monocular Road Scene Layout Estimation via Cross-View Transformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 15536–15545 (2021)
55. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3DSSD: Point-based 3D Single Stage Object Detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11040–11048 (2020)
56. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3D Object Detection and Tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11784–11793 (2021)
57. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9759–9768 (2020)
58. Zhang, Y., Lu, J., Zhou, J.: Objects are Different: Flexible Monocular 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3289–3298 (2021)
59. Zhou, X., Wang, D., Krähenbühl, P.: Objects as Points. arXiv preprint arXiv:1904.07850 (2019)
60. Zhou, Y., Tuzel, O.: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)
61. Zhou, Y., He, Y., Zhu, H., Wang, C., Li, H., Jiang, Q.: Monocular 3D Object Detection: An Extrinsic Parameter Free Approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7556–7566 (2021)
62. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. arXiv preprint arXiv:1908.09492 (2019)

63. Zhu, X., Ma, Y., Wang, T., Xu, Y., Shi, J., Lin, D.: SSN: Shape Signature Networks for Multi-class Object Detection from Point Clouds. In: European Conference on Computer Vision. pp. 581–597. Springer (2020)
64. Zou, Z., Ye, X., Du, L., Cheng, X., Tan, X., Zhang, L., Feng, J., Xue, X., Ding, E.: The Devil Is in the Task: Exploiting Reciprocal Appearance-Localization Features for Monocular 3D Object Detection. In: Proceedings of the International Conference on Computer Vision. pp. 2713–2722 (2021)